

# TANGOW: Un Sistema de Enseñanza Adaptativa a través de Internet

**Rosa María Carro, Estrella Pulido, Pilar Rodríguez**

Universidad Autónoma de Madrid, Escuela Técnica Superior de Informática

**Resumen:** En este artículo se describe TANGOW, *Task-based Adaptive learner Guidance On the Web*, un sistema para la enseñanza de cursos accesibles a través de Internet. Los cursos definidos con TANGOW se adaptan a los estudiantes teniendo en cuenta tanto sus características propias (edad, idioma, etc.), como el conjunto de acciones que realizan durante el proceso de aprendizaje. Existe una estructura asociada con cada estudiante en la que se almacena el itinerario seguido por el estudiante en su interacción con el sistema, y que se restaura al inicio de cada sesión.

Los cursos gestionados por el sistema se definen en términos de Tareas Docentes y Reglas. Las Reglas especifican la(s) relación(es) entre Tareas que, a su vez, corresponden a unidades conceptuales definidas por el diseñador del curso. Los ejemplos que se utilizan para ilustrar las características del sistema forman parte de un curso sobre educación vial.

**Palabras clave:** Aprendizaje a distancia, Multimedia e Hipermedia en educación.

## 1. Introducción

Prácticamente desde sus comienzos, tanto la red Internet como los servicios asociados a la World Wide Web han sido ampliamente utilizados para la distribución de información. Las ventajas que ambos presentan los convierten en recursos muy apropiados para el aprendizaje y la enseñanza: por una parte, las conexiones a Internet son cada vez más asequibles y, por otra, la información publicada en la Web es fácilmente accesible, con independencia del lugar y momento en que se realice el acceso, y sin necesidad de disponer de herramientas especiales para el acceso, ya que basta con utilizar un navegador estándar. Las posibles limitaciones existentes en relación al ancho de banda de la red, que pueden provocar cuellos de botella, se espera queden resueltas con Internet 2.

Estas características hacen de Internet una plataforma adecuada para los sistemas de aprendizaje *online*, pero no garantizan su efectividad, ya que para que un sistema de estas características sea útil en la enseñanza, hay que considerar otros factores que no están directamente relacionados con el soporte físico de la información, tales como el diseño de los objetos de estudio, la guía ofrecida a los estudiantes durante el proceso de aprendizaje y las relaciones entre distintos usuarios del sistema [Ausserhofer 1999].

En lo que respecta a la información que se presenta a los estudiantes, y cuyo aprendizaje constituye el objetivo del sistema, resulta conveniente que pueda ser presentada de distintas formas, adaptándose a algunas de las características propias de cada estudiante. Esto sólo será posible si, tanto la estructura de la información, como la selección de los elementos multimedia asociados, es diseñada de forma apropiada. También es importante dotar al sistema de un procedimiento de mantenimiento que permita una posible evolución de los contenidos ofrecidos durante el proceso de aprendizaje. Estos dos factores ponen en

evidencia que la tarea del diseñador de cursos debe estar incluida en el marco de cualquier sistema de enseñanza basado en Internet. Por otra parte, si se desea que el sistema pueda evolucionar en el tiempo, se deben tener en cuenta no sólo las relaciones entre cada usuario y el propio sistema, sino también las relaciones entre usuarios, que podrían potenciar la utilidad del procedimiento de enseñanza. En este sentido, debería ser posible la interacción entre estudiantes en términos de colaboración, así como la colaboración de un tutor con sus estudiantes [Hmelo 1998] [Nicol 1999].

El resto del trabajo se estructura como sigue. En la sección 2 se detalla la arquitectura de TANGOW, *Task-based Adaptive learNer Guidance On the Web*, un sistema para la creación y enseñanza de cursos adaptativos accesibles a través de Internet. La sección 3 describe las *tareas y reglas docentes* en que está basada la definición de los cursos gestionados por el sistema, mientras que la sección 4 relaciona algunos aspectos significativos del sistema propuesto con otros sistemas existentes. Por último se presenta un apartado de conclusiones, que describe el trabajo en curso en torno al sistema. Más detalles sobre el funcionamiento de TANGOW, el desarrollo de cursos, etc. se pueden encontrar en [Carro et al. 1999].

## 2. La arquitectura de TANGOW

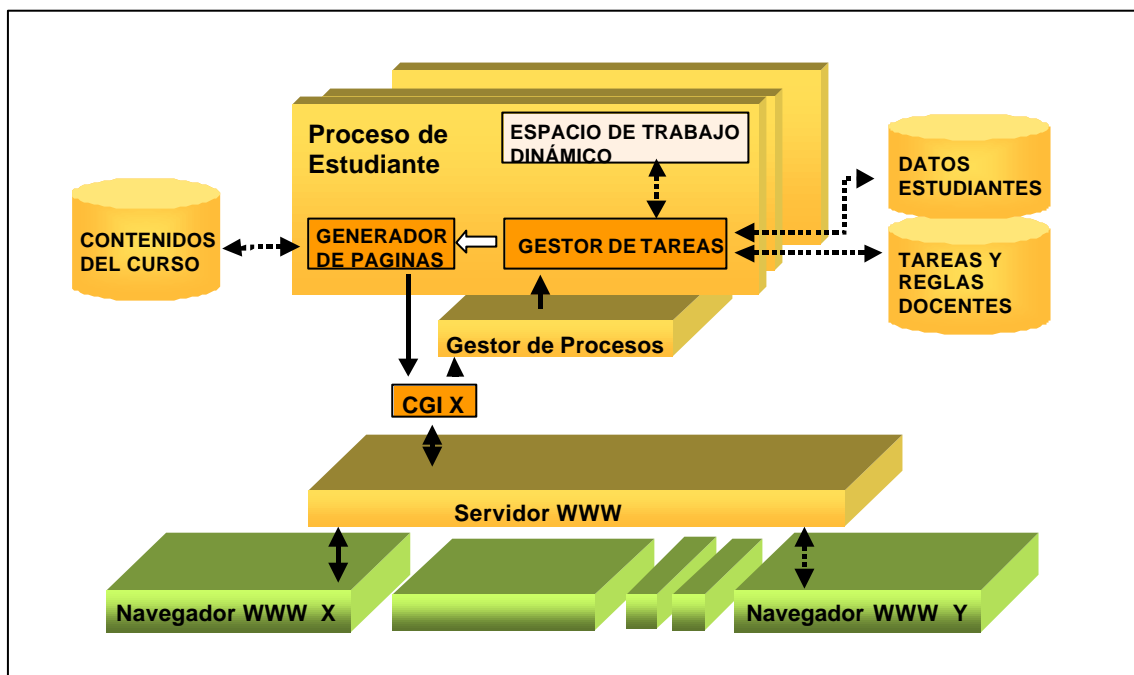
La arquitectura del sistema TANGOW está basada en el modelo Web estándar, en el que el servidor recibe peticiones de los estudiantes a través de los navegadores. Hay un proceso por cada estudiante conectado al sistema, que controla el aprendizaje del estudiante durante toda la sesión. Si un mismo estudiante está siguiendo más de un curso, entonces existirá un proceso por cada uno de los cursos. El conjunto principal de programas y datos utilizados por el sistema aparecen en la figura 1, donde las líneas de puntos representan flujo de información y las líneas continuas representan comunicación entre procesos. La flecha blanca representa una llamada a función.

### 2.1. Los programas

Los principales módulos del sistema son el *Gestor de Procesos*, el *Gestor de Tareas* y el *Generador de Páginas*. Todos ellos se encuentran situados en el servidor al que se conectan los estudiantes, y son accesibles a través de un programa CGI. La comunicación entre procesos se realiza actualmente mediante "sockets". Para que el sistema esté operativo, el *Gestor de Procesos* debe estar siempre activo, esperando peticiones procedentes de los estudiantes. El funcionamiento de los otros dos módulos dependerá de dichas peticiones. Cuando el CGI recibe una petición del cliente, envía los parámetros recibidos al *Gestor de Procesos* y se mantiene a la espera de una respuesta por parte del *Generador de Páginas*. Cuando recibe las páginas HTML generadas, se las envía al estudiante. Un ejemplo de este proceso es el correspondiente al navegador X de la figura 1. Hay que observar que las peticiones de páginas HTML estáticas no se gestionan a través del programa CGI (véase el navegador Y en la misma figura).

Si los parámetros de la petición recibidos por el *Gestor de Procesos* corresponden al comienzo de una sesión, éste lanza un nuevo *Gestor de Tareas* y establece la comunicación con él a través de un nuevo "socket". Si la petición correspondiera a una sesión ya iniciada, el *Gestor de Procesos* envía directamente los parámetros al *Gestor de Tareas* correspondiente. El principal objetivo de dicho *Gestor de Tareas* es guiar a los estudiantes en su proceso de aprendizaje, decidiendo el siguiente conjunto de tareas "realizables". Los elementos de este

conjunto dependen de la estrategia de aprendizaje activa (que ha sido previamente seleccionada por el estudiante), de los datos personales del estudiante y de sus acciones previas. Tanto la estrategia preferida por el estudiante para el aprendizaje como los datos personales del mismo son obtenidos a través de un test que el estudiante realiza la primera vez que accede a un curso, y son almacenados por el sistema para su uso tanto en la sesión actual como en sesiones posteriores. El *Gestor de Tareas* almacena información sobre las acciones llevadas a cabo por el estudiante y sus resultados (el número de páginas visitadas, el número de ejercicios hechos, el número de ejercicios resueltos correctamente, etc.) en el *Espacio de Trabajo Dinámico* en lo que se conoce como *árbol dinámico de tareas*. En este árbol, los nodos corresponden a las tareas iniciadas (o ya realizadas) por el estudiante durante el proceso de aprendizaje, mientras que los arcos representan la relación de composición existente entre dichas tareas.



**Figura 1:** Arquitectura del sistema TANGOW

Finalmente, el *Gestor de Tareas* envía al *Generador de Páginas* los parámetros asociados con la tarea activa, que se utilizarán para generar dinámicamente las páginas HTML que recibirá el estudiante. Dichos parámetros están relacionados con el perfil y con las acciones del estudiante. Basándose en esta información, el *Generador de Páginas* decide qué elementos multimedia (textos, imágenes, vídeos, animaciones, simulaciones, applets, etc.) aparecerán en el documento HTML y cómo se distribuirán en la página. La información sobre el perfil del estudiante es utilizada para seleccionar los elementos multimedia específicos más apropiados. Una vez generadas las páginas HTML, se envían al estudiante a través del programa CGI.

## 2.2. Los datos

Los módulos anteriores utilizan la información almacenada en la base de *Datos de Estudiantes*, la base de *Tareas y Reglas Docentes* y el conjunto de *Contenidos del Curso*.

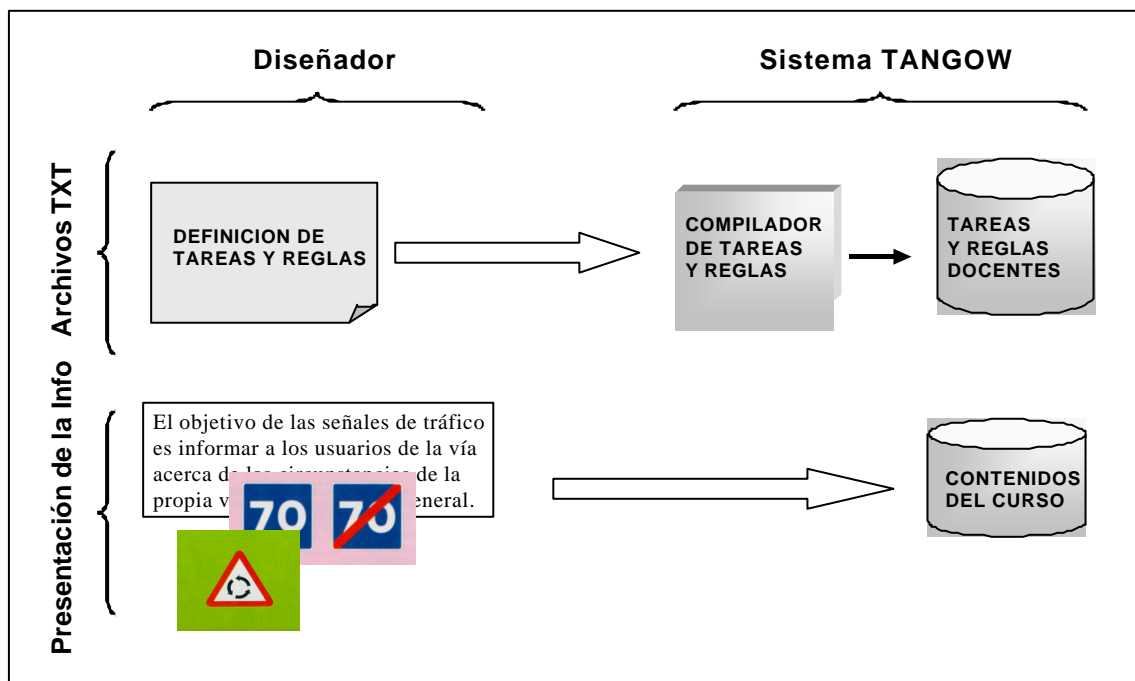
La base de *Datos de Estudiantes* contiene datos sobre los perfiles de los estudiantes y sobre sus acciones durante el proceso de aprendizaje. El perfil de estudiante incluye información personal como su edad, idioma seleccionado y preferencias con respecto a la estrategia de aprendizaje. Al final de una sesión el árbol dinámico correspondiente (parte del espacio de trabajo dinámico) se almacena también en esta base de datos.

Por otra parte, en la región reservada a los *Contenidos del Curso* se almacenan todos los elementos multimedia que aparecerán en las páginas HTML. Se clasifican de acuerdo a características definidas en el perfil de usuario.

Finalmente, la base de datos de *Tareas y Reglas Docentes* contiene la definición de todas las tareas docentes que han sido especificadas por el diseñador del curso, junto con la definición de las reglas que establecen las relaciones entre tareas.

### 3. El diseño de los cursos

Para crear un nuevo curso, el diseñador debe definir las *tareas* y *reglas* correspondientes al mismo, y clasificar los elementos multimedia que aparecerán en las páginas HTML generadas. En la versión actual del sistema, las definiciones de tareas y reglas se escriben directamente en archivos de texto, que son procesados por un compilador, el cual genera automáticamente los objetos que componen el curso. Este proceso se ilustra en la [Fig. 2].



**Figura 2:** Descripción de tareas y reglas y generación de objetos

#### 3.1 Las Tareas docentes

Una *Tarea Docente* es la unidad básica del proceso de aprendizaje, pudiendo ser *atómica* o *compuesta*. El curso completo puede considerarse como una tarea a realizar, que está

conceptualmente dividida en subtareas, las cuales pueden estar a su vez divididas en otras subtareas, y así sucesivamente, de forma que los estudiantes deben realizar todas o sólo algunas de ellas para que pueda considerarse que han realizado la tarea principal.

Respecto a su objetivo principal, cada tarea puede ser de *teoría*, de *prácticas* o de *ejemplos*, pudiendo tener asociados ciertos requisitos de finalización, implementados mediante un método que decide si la tarea ha finalizado o no. Este método puede recibir parámetros relacionados o bien con las acciones del estudiante durante la realización de esta tarea (en el caso de que la tarea sea atómica), o bien con la finalización de las subtareas (si la tarea es compuesta). Una tarea también puede tener asociada, opcionalmente, una lista de elementos multimedia que se utilizarán para la generación dinámica de las páginas HTML. Existe un lenguaje de descripción que especifica qué tipo de elementos aparecerán en cada página y sus posiciones relativas, y que se utiliza para construir las páginas que se presentan al estudiante.

Los elementos multimedia se encuentran almacenados y clasificados dependiendo de su naturaleza (p.e. nivel de dificultad o idioma). El sistema selecciona los elementos más apropiados para cada estudiante durante la creación de las páginas HTML asociadas a la tarea que está realizando el estudiante en cada momento.

### 3. 2. La relación entre Tareas Docentes

La descomposición de tareas está representada por medio de *Reglas*. Cada regla tiene un nombre e incluye información sobre la tarea compuesta, la lista de subtareas que forman parte de la misma y una palabra clave que indica la secuencia de las subtareas: puede ser necesario realizarlas todas en un determinado orden (AND), en cualquier orden (ANY), o puede ser suficiente con realizar alguna de ellas (OR o XOR). Además, una regla especifica los requisitos para su activación, que pueden depender de parámetros relacionados con las tareas ya realizadas, el perfil del usuario y/o la estrategia que está utilizándose. Por último, se indica cómo se calculan los valores de los parámetros dinámicos (número de páginas visitadas, número de ejercicios resueltos, porcentaje de aciertos, etc.) de la tarea compuesta a partir de los valores de los parámetros dinámicos de las subtareas.

### 3. 3. Ejemplos de Tareas y Reglas

En la dirección <http://www.ii.uam.es/~rcarro/Webnet99/TaskTree.html> se encuentra accesible la definición completa de tareas y reglas docentes de un curso sobre educación vial que se ha desarrollado como demostración. En dicho esquema, cada celda se corresponde con una tarea. Aquellas reglas en las que la tarea aparece en la parte izquierda de la regla se sitúan encima del nombre de la tarea. Las reglas en que la tarea aparece en la parte derecha se sitúan debajo de su nombre. La definición de las reglas que rigen el curso es independiente del idioma, así como la de tareas, con excepción, en la versión actual del sistema, del campo "descripción", que se utiliza directamente para la creación dinámica de las páginas HTML.

Por ejemplo, la tarea denominada 'SEÑALES\_CIRCULARES' [Fig. 3] es un ejemplo de una tarea de *teoría atómica*. La función asociada a sus requisitos de finalización ('f\_teo') recibe como parámetros de entrada el número de páginas visitadas por el estudiante ('pag\_visited') y el número total de páginas asociadas a la tarea ('tot\_pag'). Los elementos multimedia utilizados para generar las páginas HTML asociadas a esta tarea aparecen como valores del campo HTML. El valor M1 indica que estos elementos aparecerán uno debajo de otro.

TYPE =	T
ATOMIC =	Y
DESCRIPTION =	Descripción de las Señales Circulares
END_METHOD =	F_TEO
PARAMS =	pages_visited tot_pages
HTML =	CIRCULARES      M1 PARAR              M1 C_PROHI           M1 E_PROHI           M1 EP_VEHl           M1 EPV_LADO         M1

**Figura 3:** Descripción de la tarea docente ‘SEÑALES\_CIRCULARES’

Otro ejemplo lo constituye la tarea 'AGENTES\_DE\_TRÁFICO\_EJERCICIOS', que corresponde a una tarea *práctica* que contiene ejercicios relacionados con los agentes de tráfico [Fig. 4]. Esta tarea también es atómica. Los parámetros de entrada para el método asociado a las condiciones de finalización ('f\_prac1') son el número de ejercicios realizados por el estudiante ('exer\_done'), el número de ejercicios resueltos correctamente ('exer\_ok') y el número total de ejercicios de esta tarea ('tot\_exer'). Los elementos multimedia, así como la respuesta correcta para cada ejercicio, están especificados en los valores asignados al campo HTML. Como en el ejemplo anterior, el valor M1 para el modo indica que todos los elementos aparecerán colocados uno debajo de otro.

TYPE =	P
ATOMIC =	Y
DESCRIPTION =	Agentes de Tráfico: Ejercicios
END_METHOD =	F_PRAC1
PARAMS =	exer_ok exer_done tot_exer
HTML =	eA1      a      M1 eA2      b      M1 eA3      c      M1 eA4      b      M1 eA5      a      M1

**Figura 4:** Descripción de la tarea docente 'AGENTES\_DE\_TRÁFICO\_EJERCICIOS'

Respecto a las reglas docentes, en la [Fig. 5] se puede ver un ejemplo de definición, que corresponde a la regla denominada ‘R1’, cuya precondition viene dada por 'c\_4', lo que indica que la regla sólo se activará en caso de que la ejecución del método 'c\_4' devuelva verdadero. Este método recibe como parámetro de entrada el valor del parámetro 'exer\_ok' de la tarea ‘SEÑALES\_VERTICALES’. La propagación de parámetros queda descrita en el campo

‘CALC\_PARAMS’, donde debe aparecer, para cada parámetro cuyo valor se desea calcular, su nombre, el método a utilizar para el cálculo y una lista con los parámetros de los que depende el cálculo, junto con las tareas donde se deberán buscar sus valores. En nuestro ejemplo, el valor del parámetro ‘time\_in’ de la tarea ‘PRIORITY’ se calcula como suma de los valores de los parámetros ‘time\_in’ de las tareas ‘SEÑALES\_CIRCUNST\_TEORÍA’ y ‘SEÑALES\_CIRCUNST\_EJERC’.

SEQUENCING =	AND			
LHS =	Priority			
RHS =	SEÑALES_CIRCUNST_TEORÍA SEÑALES_CIRCUNST_EJERC			
ACT_CONDITION =	c-4			
PARAMS =	exer_ok	SEÑALES_VERTICALES		
CALC_PARAMS =	time_in	msum2	time_in	SEÑALES_CIRCUNST_TEORÍA
			time_in	SEÑALES_CIRCUNST_EJER
	tot_pag	mdirect	tot_pag	SEÑALES_CIRCUNST_TEORÍA
	exer_ok	mdirect	exer_ok	SEÑALES_CIRCUNST_EJER
	exer_done	mdirect	exer_done	SEÑALES_CIRCUNST_EJER
	tot_exer	mdirect	tot_exer	SEÑALES_CIRCUNST_EJER
	pag_visited	mdirect	pag_visited	SEÑALES_CIRCUNST_TEORÍA

**Figura 5:** Ejemplo de definición de una regla

#### 4. Otros sistemas relacionados

De especial interés en el campo de sistemas de enseñanza adaptativos es el trabajo desarrollado por el grupo ELM en los últimos años [Brusilovsky et al. 1996] [Brusilovsky & Anderson 1998]. Una de sus implementaciones es ELM-ART II [Weber & Specht 1997], un sistema tutor adaptativo para la Web, cuyo objetivo es enseñar a programar en LISP. Este sistema recibió el European Academic Software Award en 1998. ELM-ART II es adaptativo porque se elige el siguiente mejor paso en el curriculum presentado al estudiante. Los enlaces de las páginas HTML se anotan siguiendo la metáfora del semáforo, que usa distintos colores para indicar qué secciones se recomienda estudiar, cuáles se han aprendido ya o cuáles el estudiante todavía no se encuentra en condiciones de abordar. Este proceso de anotación se lleva a cabo siempre que se finaliza una unidad de aprendizaje revisando todos los conceptos que son prerequisites para esa unidad. Esta es una diferencia con TANGOW, donde se usan los árboles dinámicos para restringir el conjunto de tareas docentes que debe revisarse cuando se finaliza una tarea.

En cuanto al proceso de generación dinámica de páginas, puede mencionarse el sistema AHA [de Bra & Calvi 1998] en el que se filtran los contenidos de las páginas mediante sentencias condicionales. Estas sentencias son incluidas como comentarios en las páginas HTML. La principal diferencia con TANGOW es que, en este último, las páginas HTML son creadas desde cero, enlazando los elementos multimedia más apropiados para cada estudiante, mientras que en AHA las páginas ya están previamente creadas y se decide qué porciones de ellas se muestran y cuáles no.

Existen también otras formas de estructurar los contenidos del curso. En TANGOW se estructuran mediante tareas y reglas docentes, como se mencionó anteriormente. En [da Silva et al. 1998] se propone la estructuración del dominio en conceptos, que están asociados directamente a documentos y a otros conceptos mediante enlaces dotados de un tipo y un peso específicos. Otro formalismo que se orienta en este sentido es el de los grafos de requisitos [Nykänen 1997], que descompone el curso en celdas, cada una de ellas asociada a una unidad temática. La relación entre celdas se establece mediante el mencionado grafo de requisitos. Este último formalismo difiere de TANGOW en que cada celda tiene asociado un documento estático, siendo también estática la relación entre celdas establecida en el grafo. En TANGOW los documentos asociados a cada tarea son generados dinámicamente y los requisitos para la activación de una regla pueden variar en función del perfil del estudiante, las acciones que haya realizado hasta el momento y la estrategia de enseñanza en uso.

Otro sistema de enseñanza a través de Internet que merece la pena mencionar es SKILL [Neumann & Zirvas 1998]. En SKILL el curso se estructura de forma similar a los modelos de grafo de requisitos anteriores, pero la adaptatividad del sistema se implementa teniendo en cuenta los conocimientos previos de cada estudiante. En TANGOW no sólo se puede tener en cuenta este factor, sino también otras características propias de cada perfil, como son el lenguaje elegido para el curso o la edad del estudiante, que puede condicionar el tipo de documentación que se le ofrezca. Por otra parte, SKILL, al igual que TANGOW, almacena los resultados de cada sesión, que también se utilizarán en sesiones posteriores. Una de las principales diferencias entre SKILL y otros sistemas es que éste incorpora la posibilidad de realizar lo que se denomina “anotaciones”, que posibilitan la realización de algunos tipos de actividades de colaboración entre usuarios del sistema.

Una aproximación diferente es la propuesta en el sistema DCG [Vassileva 1997] [Vassileva 1998], donde los contenidos se estructuran como un mapa de carreteras, que se utiliza para generar el plan del curso. Para ello, el planificador busca aquellos subgrafos que conecten los conceptos ya conocidos por el estudiante con el concepto objetivo del curso, pudiendo variar la trayectoria prevista si el estudiante no alcanza el nivel prefijado respecto a al aprendizaje de ciertos conceptos. La principal diferencia entre DCG y TANGOW es que en el primero, cada vez que es necesario adaptar el curso a las características del estudiante, debe revisarse todo el plan, mientras que en TANGOW dicha adaptación se realiza en cada paso del proceso, y sólo implica la selección de un conjunto de tareas activables en cada momento.

Por último, mencionar el proyecto Benchmarks [Brown & Hansen 1998] que, aunque no propone ningún sistema nuevo, estudia uno de los principales problemas que surgen a la hora de desarrollar cursos en Web: la falta de experiencia y tiempo de dedicación de los diseñadores de los cursos. El objetivo de este proyecto es el análisis y evaluación de varios tipos de herramientas, tanto para la creación de los elementos multimedia, como para facilitar la comunicación e interacción entre diseñadores. Desde el punto de vista de los diseñadores, TANGOW facilita la creación de los cursos, ya que permite definir la estructura de las tareas o conceptos sin necesidad de asociarlos directamente a los contenidos que se presentarán al estudiante en cada caso. La organización independiente de ambos tipos de información simplifica no sólo el diseño, sino también el mantenimiento de los cursos.

## **5. Conclusiones y trabajo en curso**

El sistema TANGOW permite desarrollar cursos adaptativos para entornos Web, mediante la descripción de tareas y reglas docentes, que se utilizan, en tiempo de ejecución, para guiar a



los estudiantes durante el proceso de aprendizaje, de forma que a cada estudiante se le presentará un conjunto diferente de páginas HTML en función de su propio perfil, así como de las acciones que haya realizado previamente y de la estrategia de aprendizaje activa. Por otra parte, las páginas HTML se generan dinámicamente a partir de la información multimedia almacenada en relación a cada conjunto de tareas.

TANGOW está escrito en Java, y es accesible a través de Internet mediante cualquier navegador estándar. Gracias al almacenamiento de las tareas, las reglas y el material multimedia en bases de datos, el coste de mantenimiento de los cursos es bajo, dado que los diseñadores pueden variar, añadir o eliminar componentes del curso muy fácilmente. Por otra parte, el almacenamiento en bases de datos también permite la reutilización de componentes en diferentes cursos.

Actualmente se está trabajando en la mejora del proceso de los mecanismos de adaptación, incluyendo en el sistema la posibilidad de cambiar la estrategia de aprendizaje activa de forma dinámica, en función de los resultados obtenidos por el estudiante. También está prevista la realización de un módulo específico para el diseño de cursos, que se relacione directamente con las bases de datos, y facilite las labores de creación y mantenimiento de tareas, reglas y contenidos multimedia.

Más a corto plazo, se prevé el diseño de algunos cursos que puedan ser puestos a disposición de estudiantes de Ingeniería Informática, y cuya experimentación permitirá realizar una evaluación concreta de la efectividad del sistema.

Por otra parte, pensamos que la arquitectura de TANGOW es apropiada para incluir facilidades de colaboración entre los usuarios del sistema, mediante el establecimiento de procedimientos controlados de comunicación entre los procesos de los estudiantes. Ésta es otra línea de investigación pendiente de abordar en un futuro próximo.

## **Agradecimientos**

Este trabajo ha sido realizado dentro del proyecto InterEdu, financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT), proyecto número TEL97-0306.

## **Bibliografía**

Ausserhofer, A. (1999). Web-Based Teaching and Learning: A Panacea?, *IEEE Communications*, Volume 37, Number 3, pp. 92-96, March 1999.

de Bra, P. & Calvi, L. (1998). AHA: A Generic Adaptive Hypermedia System, *Second Workshop on Adaptive Hypertext and Hypermedia*, at the *Ninth ACM Conference on Hypertext and Hypermedia*, pp. 5-11. Pittsburgh, USA, June 20-24, 1998.

Brown, A. & Hansen, L. (1998). Software Tools for Distance Learning: The Benchmarks Project, *SITE 98 in ED-MEDIA 98 Conference of the AACE*, Washington DC, March 10-14, 1998.

Brusilovsky, P., Schwarz, E. & Weber, G. (1996). A tool for Developing Adaptive Electronic Textbooks on WWW, *WebNet-96 of the AACE*, Boston, 1996.  
<http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html>.

Brusilovsky, P. & Anderson, J. (1998). ACT-R electronic bookshelf: An adaptive system for learning cognitive psychology on the Web, *WebNet 98 World Conference of the WWW, Internet & Intranet*, pp. 92-97, Orlando, Florida, November 7-12, 1998.

Carro, R.M., Pulido, E. & Rodríguez, P. (1999). TANGOW: Task-based Adaptive learner Guidance On the Web, *Second Workshop on Adaptive Systems and User Modeling on the Web at the Eight International World Wide Web Conference*. Toronto, Canada, May 11-14, 1999.

Hmelo, C., Guzdial, M. & Turns, J. (1998). Computer-Support for Collaborative Learning: Learning to Support Student Engagement, *AACE Journal of Interactive Learning Research (JILR)*, Volume 9, Number 2, p. 107, 1998.

Neumann, G. & Zirvas, J.(1998). SKILL: A Scalable Internet-Based Teaching and Learning System, *WebNet 98 World Conference of the WWW, Internet & Intranet*, pp. 688-693. Orlando, Florida, November 7-12, 1998.

Nicol, J., Gutfreund, Y., Paschetto, J., Rush, K.& Martin,C. (1999). How the Internet Helps Build Collaborative Multimedia Applications, *Communications of the ACM*, Volume 42, Number 1, pp. 79-85, January 1999.

Nykänen, O.(1997). User Modeling in WWW with Prerequisite Graph Model, *Workshop on Adaptive Systems and User Modeling on the World Wide Web in the Sixth International Conference on User Modeling*, Chia Laguna, Sardinia, June 2-5 1997.

da Silva, D.P, Van Durm, R., Duval, E. & Olivi, H. (1998). Concepts and documents for adaptive educational hypermedia: a model and a prototype, *Second Workshop on Adaptive Hypertext and Hypermedia*, pp. 35-43. Ninth ACM Conference on Hypertext and Hypermedia, Pittsburgh, USA, June 20-24, 1998.

Vassileva, J. (1997). Dynamic Courseware Generation on the WWW, *8th World Conference of the AIED Society*, pp. 498-505, Kobe, Japan, August 1997.  
[http://www.contrib.andrew.cmu.edu/~plb/AIED97\\_workshop/Vassileva/Vassileva.html](http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Vassileva/Vassileva.html)

Vassileva, J. (1998). A Task-Centred Approach for User Modeling in a Hypermedia Office Documentation System, in Brusilovsky, P., Kobsa, A. and Vassileva J. (Eds.) *Adaptive Hypertext and Hypermedia*, Kluwer Academic Publ. Dordrecht, Chapter 8, pp. 209-247, 1998.

Weber, G & Specht, M. (1997). User modeling and adaptive navigation support in WWW-based tutoring systems, *User Modeling '97*, pp. 289-300, Italy, June, 1997.

### **Dirección de los autores**

Rosa María Carro Salas ([Rosa.Carro@ii.uam.es](mailto:Rosa.Carro@ii.uam.es)) Tel. 91 348 22 91

Estrella Pulido Cabañete ([Estrella.Pulido@ii.uam.es](mailto:Estrella.Pulido@ii.uam.es)) Tel. 91 348 22 85

Pilar Rodríguez Marín ([Pilar.Rodríguez@ii.uam.es](mailto:Pilar.Rodríguez@ii.uam.es)) Tel. 91 348 22 83

Universidad Autónoma de Madrid, Escuela Técnica Superior de Informática; Ctra. Colmenar Viejo, Km. 15; 28049 Madrid. Fax: 91 348 22 35.